COLA: CONSISTENT LEARNING WITH OPPONENT-LEARNING AWARENESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Optimization problems with multiple, interdependent losses, such as Generative Adversarial Networks (GANs) or multi-agent RL, are commonly formalized as *differentiable games*. Learning with Opponent-Learning Awareness (LOLA) introduced *opponent shaping* to this setting. More specifically, LOLA introduced an augmented learning rule that accounts for the agent's influence on the anticipated learning step of the other agents. However, the original LOLA formulation is *inconsistent* because LOLA models other agents as *naive learners* rather than LOLA agents. In previous work, this inconsistency was stated to be the root cause of LOLA's failure to preserve stable fixed points (SFPs). We provide a counterexample by investigating cases where Higher-Order LOLA (HOLA) converges. Furthermore, we show that, contrary to claims made, Competitive Gradient Descent (CGD) does not solve the consistency problem. Next, we propose a new method called Consistent LOLA (COLA), which *learns* update functions that are consistent under mutual opponent shaping. Lastly, we empirically compare the performance and consistency of HOLA, LOLA and COLA on a set of general-sum learning games.

1 INTRODUCTION

Multi-objective problems can be found in many domains, such as GANs (Goodfellow et al., 2014) or single- and multi-agent reinforcement learning (RL) in the form of imaginative agents (Racanière et al., 2017), hierarchical RL (Barto & Mahadevan, 2002), and intrinsic curiosity (Schmidhuber, 1991). A popular framework to understand systems with multiple, interdependent losses is differentiable games (Balduzzi et al., 2018). For example, in the case of GANs, the differentiable game framework models the generator and the discriminator as competing agents, each trying to optimize their respective loss. The action space of the game consists of choosing the respective network parameters (Balduzzi et al., 2018).

An effective paradigm to improve learning in differentiable games is *opponent shaping*, where the players use their ability to shape each other's learning steps. LOLA (Foerster et al., 2018) was the first work to make explicit use of opponent shaping in the differentiable game setting. LOLA is also one of the only general learning methods designed for differentiable games that obtains mutual cooperation with the Tit-for-Tat strategy in the Iterated Prisoner's Dilemma (IPD). The Tit-for-Tat strategy starts out cooperating and retaliates once whenever the opponent does not cooperate. It achieves mutual cooperation and has proven to be successful at IPD tournaments (Axelrod, 1984; Harper et al., 2017). In contrast, naive gradient descent and other more sophisticated methods typically converge to the mutual defection policy under random initialization (Letcher et al., 2019b).

While LOLA discovers these interesting equilibria, the original LOLA formulation is inconsistent because LOLA agents assume that their opponent is a *naive learner*. This assumption is clearly violated if two LOLA agents learn together in a game. Up until now, it was believed that this inconsistency is the root cause for LOLA's shortcomings, such as not converging to SFPs in some simple quadratic games (Letcher et al., 2019b).

How can LOLA's inconsistency be resolved? To answer this question, we first revisit the concept of *higher-order* LOLA (HOLA) (Foerster et al., 2018). For example, *second-order* LOLA assumes that the opponent is a *first-order* LOLA agent (which in turn assumes the opponent is a naive learner)

and so on. Assuming that HOLA converges with increasing order, we define *infinite-order* LOLA (iLOLA) as the limit of HOLA.

Intuitively, it should follow that two iLOLA agents have a consistent view of each other, meaning they make an accurate assumption about the learning behavior of the opponent under mutual opponent shaping.

Previous work claimed that a series expansion of Competitive Gradient Descent (CGD) (Schäfer & Anandkumar, 2020) corresponds to high-order LOLA, which implies that exact CGD corresponds to iLOLA and thus solves the consistency problem.

Contributions First, we show that, contrary to claims made, CGD does not correspond to high-order LOLA and thus cannot correspond to iLOLA either. Therefore CGD does not resolve the inconsistency problem. We continue by investigating HOLA's convergence. We introduce a formal definition of consistency and show that if HOLA converges, then the update in the limit, iLOLA, is self-consistent under mutual opponent shaping. However, we demonstrate empirically that, contrary to popular belief, resolving this inconsistency problem does not resolve LOLA's convergence issues (maintaining the original SFPs).

Second, we propose Consistent LOLA (COLA) as an alternative to addressing the inconsistency problem. Instead of repeatedly applying the LOLA learning rule, like in iLOLA, COLA learns a pair of consistent update functions by explicitly minimizing a consistency loss. By reframing the problem as such, the method only requires up to second-order derivatives, and instead of having a handcrafted update function like in LOLA or CGD, we use the representation power of neural networks to learn the update step. We provide empirical evidence that COLA finds the iLOLA solution when HOLA converges but finds different, more consistent and stable solutions when HOLA diverges. We also find that optimizing for consistency increases the robustness to a wide range of learning rates. Interestingly, while COLA (unlike LOLA) does not find Tit-for-Tat on the IPD, it does learn policies with near-optimal total payoff.

2 Related Work

General-sum learning algorithms and their consequences have been investigated from different perspectives in the reinforcement learning, game theory, and GAN literature, see e.g. (Schmidhuber, 1991; Barto & Mahadevan, 2002; Racanière et al., 2017; Goodfellow et al., 2014) to name a few. Next, we will highlight a few of the approaches to the mutual opponent shaping problem.

Opponent modeling maintains an explicit belief of the opponent, which allows to reason over their strategies and compute optimal responses. Opponent modeling can be divided into different subcategories: There are classification methods, classifying the opponents into pre-defined types (Weber & Mateas, 2009; Synnaeve & Bessiere, 2011), or policy reconstruction methods, where we explicitly predict the actions of the opponent (Mealing & Shapiro, 2017). Most closely related to opponent shaping is recursive reasoning, where methods model nested beliefs of the opponents (He et al., 2016; Albrecht & Stone, 2019; Wen et al., 2019).

In comparison, COLA assumes that we have access to the ground-truth model of the opponent, e.g., the opponent's payoff function, parameters, and gradients, which puts COLA into the framework of differentiable games (Balduzzi et al., 2018). Various methods have been proposed, investigating the local convergence properties to different solution concepts (Mescheder et al., 2018; Mazumdar et al., 2019; Letcher et al., 2019b; Azizian et al., 2020; Schäfer & Anandkumar, 2020; Schäfer et al., 2020; Hutter, 2020). Most of the work in differentiable games has not focused on the issue of opponent shaping and consistency. Mescheder et al. (2018) and Mazumdar et al. (2019) focus solely on zero-sum games without shaping. Letcher et al. (2019b) improve on LOLA, but do not investigate the consistency issue. CGD (Schäfer & Anandkumar, 2020) addresses the consistency issue of LOLA for zero-sum games but not for general-sum games. The exact difference between CGD and LOLA is addressed in the Section 4.1.

Recent work interprets the above methods as forming gradient conjectures about the opponent (Chasnov et al., 2020). In other words, we can classify the above methods by the guess they make about the learning rule of the opponent. However, the paper does not investigate the consequences of methods having consistent gradient conjectures.

3 BACKGROUND

3.1 DIFFERENTIABLE GAMES

The framework of differentiable games has become increasingly popular to model the problem of multi-agent learning. Whereas in the framework of stochastic games we are typically limited to parameters such as action-state probabilities, differentiable game generalizes to any parameters as long as the loss function is differentiable with respect to them (Balduzzi et al., 2018).

Definition 1. In a two-player differentiable game, players $i = 1, 2$ control parameters $\theta_i \in \mathbb{R}^{d_i}$ to minimize twice continuously differentiable losses $L^i : \mathbb{R}^{d_1+d_2} \to \mathbb{R}$.

A fundamental challenge of the multi-loss setting is finding a meaningful solution concept. Whereas in the single loss setting the typical solution concept is local minima, in multi-loss settings there are different sensible solution concepts. Most prominently, there are Nash Equilibria (Osborne & Rubinstein, 1994). However, Nash Equilibria include unstable saddle points that cannot be reasonably found via gradient-based learning algorithms (Letcher et al., 2019b). A more appropriate concept are stable fixed points (SFPs), which could be considered a differentiable game analogon to local minima in single loss optimization. We will omit a formal definition here for brevity and point the interested reader to previous work on the topic (Letcher et al., 2019a).

3.2 LOLA

Let us assume we are modeling a differentiable game with two players. A LOLA agent θ_1 uses its access to the opponents parameters θ_2 to differentiate through the learning step of the opponent. In other words, agent 1 reformulates their loss to $L^1(\theta_1, \theta_2 + \Delta\theta_2)$, where $\Delta\theta_2$ represents the assumed learning step of the opponent. In first-order LOLA we assume the opponent to be a naive learner: $\Delta\theta_2 = -\alpha\nabla_2 L^2$, which is what makes LOLA inconsistent if the opponent was any other type of learner. Note that ∇_2 denotes the gradient with respect to θ_2. Also note that α represents the *look-ahead* rate, which is the assumed learning rate of the opponent. In the original paper the loss was approximated using a Taylor expansion $L^1 + \nabla_2 L^1 \cdot \Delta\theta_2$. For agent 1, their first-order (Taylor) LOLA update is then defined as

$$\Delta\theta_1 := -\alpha\left(\nabla_1 L^1 + \left(\nabla_{21} L^1\right)^\top \Delta\theta_2 + \left(\nabla_1 \Delta\theta_2\right)^\top \nabla_2 L^1\right).$$

Alternatively, in *exact* LOLA, the derivative is taken directly with respect to $L^1(\theta_1, \theta_2 + \Delta\theta_2)$.

LOLA has had some empirical success, being one of the first general learning methods to discover Tit-for-Tat like solutions social dilemma. However, later work showed that in general LOLA does not preserve SFPs $\bar{\theta}$ as the LOLA term $\alpha\left(\nabla_{12} L^2\right)^\top \nabla_2 L^1$ can be nonzero at $\bar{\theta}$. In fact, LOLA agents show "arrogant" behavior: They assume they can shape the learning of their *naive* opponents without having to adapt to the shaping of the opponent. Prior work hypothesized that this arrogant behavior and hence the failure to preserve SFPs is due to LOLA's inconsistent formulation (Letcher et al., 2019a).

3.3 CGD

CGD (Schäfer & Anandkumar, 2020) proposes updates that are themselves Nash Equilibra of a local bilinear approximation of the game. It stands out by its robustness to different step sizes of opponents and its ability to find SFPs. However, CGD has been shown to not find Tit-for-Tat on the IPD, instead converging to mutual defection. CGDs update rule can be written as

$$\begin{pmatrix} \Delta\theta_1 \\ \Delta\theta_2 \end{pmatrix} := -\begin{pmatrix} \mathrm{Id} & \alpha\nabla_{12} L^1 \\ \alpha\nabla_{21} L^2 & \mathrm{Id} \end{pmatrix}^{-1} \begin{pmatrix} \nabla_1 L^1 \\ \nabla_2 L^2 \end{pmatrix} \tag{1}$$

One can recover different orders of CGD by approximating the inverse matrix via the series expansion $\lambda_{\max}(A) < 1 \Rightarrow (\mathrm{Id} - A)^{-1} = \lim_{N\to\infty} \sum_{k=0}^{N} A^k$, by letting

$$A := \begin{pmatrix} 0 & -\alpha\nabla_{12} L^1 \\ -\alpha\nabla_{21} L^2 & 0 \end{pmatrix}. \tag{2}$$

For example, at N=1, we recover a version called Linearized CGD (LCGD), defined via $\Delta\theta_1 := -\nabla_1 L^1 + \alpha\nabla_{12} L^1 \nabla_2 L^2$.

Table 1: (a) This table shows the log of the squared consistency loss on the Tandem game, where e.g. HOLA6 is sixth-order higher-LOLA. (b) Cosine similarity between COLA and LOLA, HOLA2, and HOLA6 over different look-ahead rates on the Tandem game.

(a)

α	LOLA	HOLA2	HOLA6	COLA
1.0	128.0	512	131072	4.84e-14
0.5	12.81	14.05	12.35	2.62e-14
0.3	2.61	2.05	0.66	4.09e-14
0.1	0.08	9.13e-3	1.62e-6	6.55e-14
0.01	1.41e-5	2.10e-8	3.69e-14	8.58e-14

(b)

α	LOLA	HOLA2	HOLA4
1.0	0.57	0.58	0.60
0.5	0.61	0.46	0.15
0.3	0.92	0.51	0.72
0.1	0.94	0.98	0.99
0.01	0.99	1.0	1.0

4 METHOD AND THEORY

4.1 CGD IS NOT EQUAL TO INFINITE-ORDER LOLA

Schäfer & Anandkumar (2020) state that the series-expansion of CGD recovers higher-order Taylor LOLA. This would imply that CGD is equal to iLOLA (i.e., the limit of computing higher and higher orders of LOLA), and thus provides a consistent opponent shaping rule.

Unfortunately, this is, in general, not true. For instance, when comparing LCGD to the previously derived LOLA update function, we are missing the typical LOLA term containing the derivative through the learning step of the opponent. LCGD and LOLA are thus only equivalent in zero-sum games. Similarly, for $N = 2$, it is

$$\sum_{k=0}^{N=2} A^k = \begin{pmatrix} \mathrm{Id} + \alpha^2 \nabla_{12} \mathrm{L}^1 \nabla_{21} \mathrm{L}^2 & -\alpha \nabla_{12} L^1 \\ -\alpha \nabla_{21} L^2 & \mathrm{Id} + \alpha^2 \nabla_{21} L^2 \nabla_{12} L^1 \end{pmatrix} \tag{3}$$

such that the second-order CGD update function turns out to be

$$\Delta\theta_1 = -\nabla_1 L^1 - \alpha^2 \nabla_{12} L^1 \nabla_{21} L^2 + \alpha \nabla_{12} L^1. \tag{4}$$

Although this includes a gradient with respect to the opponent's learning step, we still do not recover the LOLA term $\alpha \left(\nabla_{12} L^2\right)^\top \nabla_2 L^1$. One can see that continuing the expansion would also not recover the LOLA term. This is not a rigorous proof, but it should be intuitively clear that, in general, CGD will not recover higher-order and thus infinite-order LOLA. This conclusion is also supported by our empirical results (Section 6.1).

4.2 CONVERGENCE AND CONSISTENCY OF HIGHER-ORDER LOLA

Next, we turn to analyzing iLOLA. Here, and for the rest of this paper, we will focus on exact LOLA.

First, HOLA does not always converge. Even in simple quadratic games, it may not converge for high look-ahead rates (see Appendix A). We leave it to future work to investigate theoretically when it does converge, e.g., given a small enough look-ahead rate.

Second, if HOLA converges, then we can define iLOLA as the limit. Assuming the gradients with respect to the LOLA update converge as well, we can show that iLOLA is consistent. We define consistency formally as follows:

Definition 1 (Consistency)**.** Any two differentiable update functions $h_1 \colon \mathbb{R}^d \to \mathbb{R}^{d_1}$ and $h_2 \colon \mathbb{R}^d \to \mathbb{R}^{d_2}$ are consistent (under mutual opponent shaping) if for all $\theta_1 \in \mathbb{R}^{d_1}, \theta_2 \in \mathbb{R}^{d_2}$, they satisfy

$$h_1(\theta_1, \theta_2) = -\alpha \nabla_1 (L^1(\theta_1, \theta_2 + h_2(\theta_1, \theta_2))) \tag{5}$$

$$h_2(\theta_1, \theta_2) = -\alpha \nabla_2 (L^2(\theta_1 + h_1(\theta_1, \theta_2), \theta_2)) \tag{6}$$

Now let h_i^n denote player i's n-th order exact LOLA update. Then iLOLA is defined as the pointwise limit $h_i(\theta) := \lim_{n\to\infty} h_i^n$ for any $\theta \in \mathbb{R}^d$, if the limit exists. We can then show

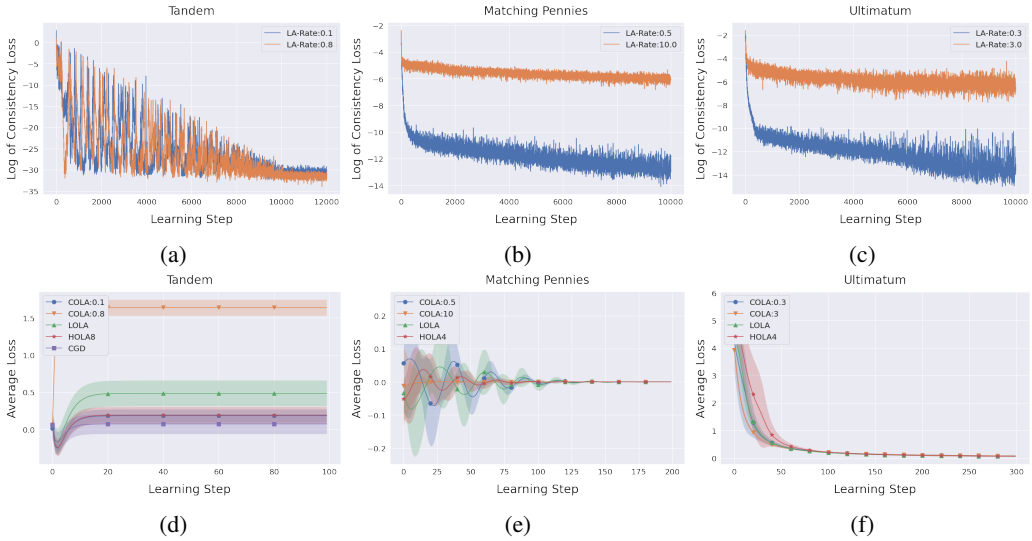


Figure 1: Subfigure (a), (b) and (c) depicts the log of the consistency loss over the training of the update functions for the Tandem, MP and Ultimatum games. Subfigure (d), (e) and (f) show the performance of COLA in comparison to HOLA, LOLA and CGD. COLA:0.1 denotes COLA with a look-ahead rate of 0.1.

Proposition 1. *Assume that for $i = 1, 2$ and any θ, it is*

$$\lim_{n\to\infty} \nabla_i h^n_{-i}(\theta) := \nabla_i h_{-i}(\theta)$$

(where $-i$ denotes the other player); i.e., in addition to h^n, also the derivatives $\nabla_i h^n_{-i}$ converge pointwise. Then the pair of update functions h_1, h_2 is consistent.

Proof. In Appendix B. □

4.3 COLA

iLOLA is consistent under mutual opponent shaping. However, HOLA does not in general converge and, even when it does, it might further be infeasible to compute HOLA to a sufficient order to achieve convergence.

As an alternative, we propose consistent LOLA (COLA). COLA finds consistent update functions by minimizing a consistency loss derived from the definition above for any pair of update functions, h_1 and h_2, parameterized by ϕ_1 and ϕ_2:

$$\Delta\theta_1 = h_1(\theta_1, \theta_2) \tag{7}$$

$$\Delta\theta_2 = h_2(\theta_1, \theta_2) \tag{8}$$

The consistency loss for h_1 for a given pair of weights θ_1, θ_2 is

$$C_1(\phi_1, \phi_2, \theta_1, \theta_2) = \left\| h_1(\theta_1, \theta_2) - \left(-\alpha \nabla_1 (L^1(\theta_1, \theta_2 + h_2(\theta_1, \theta_2)))\right) \right\|^2 \tag{9}$$

There is an equivalent equation for h_2 and when both losses are 0 the two update functions are consistent under *mutual opponent shaping*.

For this paper, we parameterize h_1 and h_2 with neural networks and numerically minimize the loss over a region of interest using Adam (Kingma & Ba, 2017).

The parameter region of interest Θ depends on the game being played. For a game with probabilities as actions, we select a parameter grid that captures most of the probability space (e.g. we sample a pair of parameters $(\theta_1, \theta_2) \sim [-7, 7]$ as $\sigma(7) \approx 1$, where σ is the Sigmoid function).

Table 2: (a) Comparison of consistency losses over multiple look-ahead rates on the MP game. (b) Cosine similarity between COLA and LOLA, HOLA2 and HOLA4 over different look-ahead rates on the MP game.

(a)

α	LOLA	HOLA2	HOLA4	COLA
10	0.06	0.70	6.56	0.24
5	4.59e-3	0.03	0.15	9.47e-3
1.0	8.79e-6	3.25e-8	4.37e-9	2.35e-7
0.5	4.80e-7	2.53e-10	5.18e-12	1.30e-7
0.01	1.07e-13	5.58e-17	5.30e-17	6.99e-8

(b)

α	LOLA	HOLA2	HOLA4
10	0.90	0.87	0.68
5	0.98	0.95	0.89
1.0	0.99	0.99	0.99
0.5	0.99	0.99	0.99
0.01	0.99	0.99	0.99

We then train ϕ_1, ϕ_2 to minimize the loss

$$C(\phi_1, \phi_2) = \mathbb{E}_{(\theta_1,\theta_2)\sim\mathcal{U}(\Theta)} \left[C_1(\phi_1, \phi_2, \theta_1, \theta_2) + C_2(\phi_1, \phi_2, \theta_1, \theta_2)\right] \tag{10}$$

via stochastic gradient descent. That is, we sample parameter pairs (θ_1, θ_2) uniformly from the grid Θ and feed them to the neural networks h^1 and h^2 respectively, each outputting the parameter update of one of the agents $(\Delta\theta_1, \Delta\theta_2)$. We then update the neural network parameters ϕ_1, ϕ_2 by taking a gradient step to minimize the sum of the consistency losses, C_1 and C_2.

We train the update functions until the loss has converged. Only after this process, these update functions are used to train a pair of agent policies.

5 Experiments

We carry out our investigation on a set of games from the literature (Balduzzi et al., 2018; Letcher et al., 2019b). For details on the training procedure of COLA, we refer the reader to Appendix C.

First, we compare HOLA and COLA on quadratic, general-sum games, including the Tandem game (Letcher et al., 2019b), where LOLA fails to converge to SFPs. Second, we investigate non-quadratic games, such as the zero-sum Matching Pennies (MP) game, the general-sum Ultimatum game (Hutter, 2020) and the iterated prisoner's dilemma (IPD) (Axelrod, 1984; Harper et al., 2017).

We investigate the convergence behavior of HOLA and COLA by comparing the consistency losses over a range of look-ahead rates, where COLA is retrained for each look-ahead rate to ensure a fair comparison. To compare the solutions found by HOLA and COLA, we compute the cosine similarity between the two across randomly sampled parameters across our region of interest.

5.1 Quadratic Games

Tandem Game. In the Tandem game (Letcher et al., 2019b), two agents sit on a tandem bike, facing opposite directions. Their actions, θ_1 and θ_2, are the forces applied to the pedals respectively. An agent can pedal backward by using negative values. The goal is to move in one direction, e.g. $\theta_1 \approx -\theta_2$, however, the agents receive a penalty for pedaling backward. This is captured in the loss

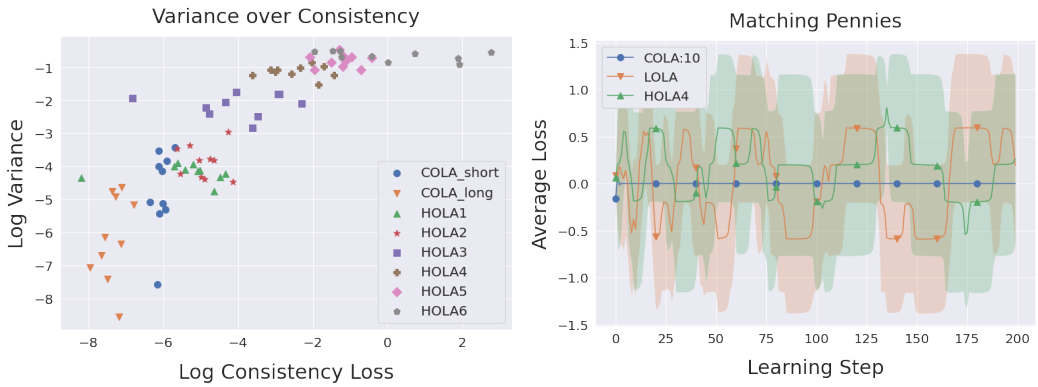
$$L^1(\theta_1, \theta_2) = (\theta_1 + \theta_2)^2 - 2\theta_1 \quad \text{and} \quad L^2(\theta_1, \theta_2) = (\theta_1 + \theta_2)^2 - 2\theta_2 \tag{11}$$

for agent 1 and 2 respectively. The Tandem game was originally introduced to show that LOLA fails to preserve SFPs at $\theta_1 + \theta_2 = 1$ and instead converges to sub-optimal solutions (Letcher et al., 2019b).

Additionally to the Tandem game, we investigate the algorithms on the quadratic Balduzzi and Hamiltonian game (Balduzzi et al., 2018). The details of the experiments are in Appendix D.

5.2 Non-Quadratic Games

Matching Pennies. Matching Pennies (MP) is a single-shot, zero-sum game, where two players, A and B, each flip a biased coin (Lee & K, 1967). Player A wins if the outcomes of both flips are the same and player B wins if they are different. The payoff matrix is shown in Table 4a. Each policy



(a) (b)

Figure 2: Training in MP at look-ahead rate of $\alpha = 10$. (a) Axes are on a log-scale. Increasing the consistency helps with decreasing the variance of the solution. (b) LOLA and HOLA find non-convergent or even diverging solutions, while COLA is stable.

is parameterized with a single parameter, the log-odds of choosing heads $p_{\text{heads}} = \sigma(\theta_A)$. In this game, the unique Nash equilibrium is playing heads half the time.

Ultimatum Game. The binary, single-shot Ultimatum game (Güth et al., 1982; Sanfey et al., 2003; Oosterbeek et al., 2004; Henrich et al., 2006) is set up as follows. There are two players, player A and B. Player A has access to $10. They can split the money fairly with B ($5 for each player) or they can split it unfairly ($8 for player A, $2 for player B). Player B can either accept or reject the proposed split. If player B rejects, the reward is 0 for both players. If player B accepts, the reward follows the proposed split. Player A's parameter is the log-odds of proposing a fair split $p_{\text{fair}} = \sigma(\theta_A)$. Player B's parameter is the log-odds of accepting the unfair split (assuming that player B always accepts fair splits) $p_{\text{accept}} = \sigma(\theta_B)$.

$$V_A = 5p_{\text{fair}} + 8(1 - p_{\text{fair}})p_{\text{accept}} \quad \text{and} \quad V_B = 5p_{\text{fair}} + 2(1 - p_{\text{fair}})p_{\text{accept}} \tag{12}$$

IPD. We next investigate the infinitely iterated prisoners' dilemma with discount factor $\gamma = 0.96$ and the usual payout function (see Appendix E). An agent i is defined through 5 parameters, the log-odds of cooperating for the first time step and for the four possible tuples of past actions of both players in the later steps.

6 Results

Our experiments aim to address a few questions: **(1)** Does HOLA always converge, and if not, under which parameter regimes does it not converge? **(2)** When HOLA converges, does COLA converge to the same solution? **(3)** Does COLA find a solution even when HOLA does not converge? **(4)** Does COLA always find the same solution in practice and which solution does COLA actually find? **(5)** What is the empirical effect of COLA on the *learning outcome* in different settings?

6.1 Quadratic Games

Table 1a provides answers to questions **(1)** - **(3)** for the quadratic games we tested on. Note that HOLA's consistency losses decrease with increasing order of HOLA for look-ahead rates below 0.5. The decrease in consistency loss with increasing order indicates that HOLA converges with increasing order. However, the consistency losses increase with look-ahead rates above 0.5. Therefore, HOLA does not always converge, even on simple quadratic games such as the Tandem game.

Table 1b shows the cosine similarities between the HOLA and COLA update functions. When HOLA converges, COLA also finds a consistent solution, and HOLA and COLA become more similar with increasing order, indicating that they find the same solution. In the cases where HOLA does not converge, COLA still finds consistent solutions. Here, the similarity scores indicate no similarity and fluctuate greatly, probably because the higher-order gradients of HOLA become increasingly

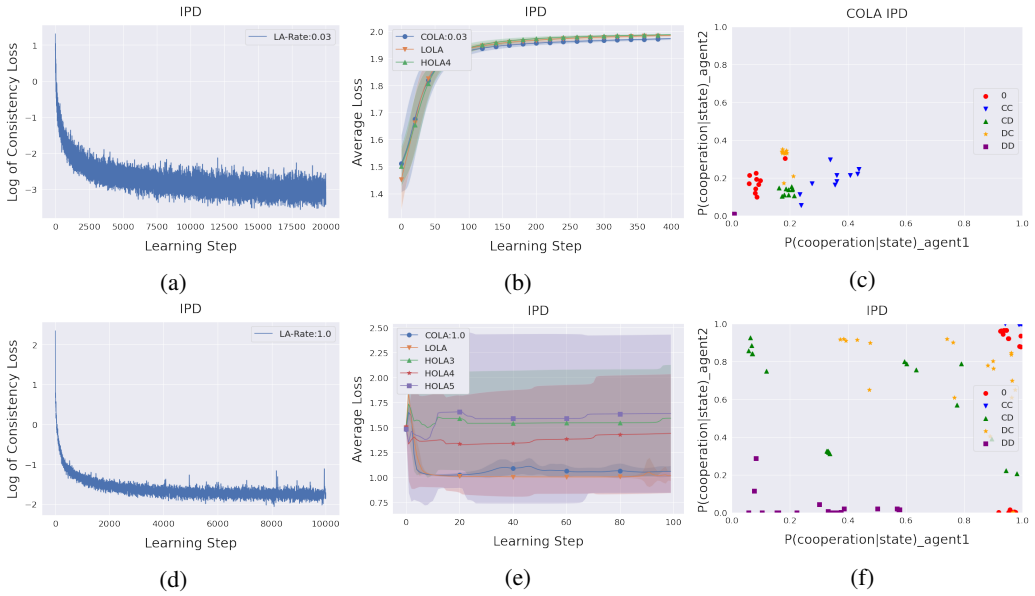


Figure 3: Results are on the IPD. Subfigure (a) / (d), show the consistency loss for look-ahead rate of 0.03 / 1.0 respectively, (b) / (e) the average loss and (c) / (f) the policy for the first player, both for the same pair of look-ahead rates. At low look-ahead HOLA defects and at high ones it diverges, also leading to high loss.

noisy. COLA still finds stable solutions in this case, as shown by the performance of COLA at a look-ahead rate of 0.8 in Figure 1d, showcasing the benefits of higher consistency.

For a qualitative comparison between HOLA and COLA, note that on Figure 1d the solution of HOLA and COLA are nearly identical in the Tandem game, given the same look-ahead rate. Also, COLA or HOLA still do not find the SFP, showing that the consistency issue was not the (only) reason for the arrogant behavior of LOLA.

Over 5 convergent COLA training runs, we see that COLA always finds very similar solutions. For more details we refer to Table 6 in the Appendix. Please note that COLA's training does not always succeed in finding the best solution at high look-ahead rates.

In Appendix D, we confirm our results on the Balduzzi and Hamiltonian games.

6.2 Non-quadratic Games

Next, we will address our questions for a set of non-quadratic games. Like with the quadratic games, we find that HOLA only converges under certain look-ahead rates and in those cases COLA finds very similar solutions. In contrast to the quadratic games, when HOLA converges, COLA's consistency loss is now higher than HOLA's. Nonetheless, the similarity scores show that COLA's solution is nearly the same. While the consistency loss for high look-ahead rates is high compared to small ones, in those cases COLA finds more consistent solutions than HOLA. The exact consistency losses are given in Table 2a and Appendix E.

In a visual comparison between the gradient fields of HOLA and COLA, we see that a diverging HOLA results in chaotic gradients, whereas COLA is still able to learn meaningful and stable update functions, which further reinforces the importance of consistency. Visualizations are given in Appendix E.

Our results show various benefits of increased consistency in MP. We compare the losses of HOLA and COLA with different look-ahead rates in MP in Figure 1e. The high look-ahead rate case is shown in Figure 2b. With small look-ahead rates COLA shows slower convergence than HOLA. While HOLA at high look-ahead rates diverges quickly, COLA is still able to find a stable solution. In Figure 2a, we plot the variance of the solutions found by different methods against their respective

Table 3: (a) Comparison of consistency losses over multiple look-ahead rates on the IPD game. (b) Cosine similarity between COLA and LOLA, HOLA2 and HOLA4 over different look-ahead rates, α on the IPD game.

(a)

α	LOLA	HOLA2	HOLA4	COLA
1.0	39.56	21.16	381.21	0.65
0.03	1.72e-3	4.72e-6	9.72e-8	0.33

(b)

α	LOLA	HOLA2	HOLA4
1.0	0.77	0.70	0.53
0.03	0.96	0.98	0.98

consistency loss, at a high look-ahead rate. The variance of the solution on the MP game decreases significantly with decreasing consistency loss.

We can also confirm empirically that COLA finds similar solutions over multiple convergent runs in Appendix E.

For more results and qualitative comparisons, we refer the reader to Appendix E.

6.3 IPD

For the IPD we look at two different look-ahead rates, one where HOLA converges and one where it diverges. Previous work investigated the IPD on look-ahead rates where HOLA diverges (Letcher et al., 2019b).

Both LOLA and COLA find the Defect-Defect strategy under the look-ahead regimes where HOLA converges, as can be seen in Figure 3. Furthermore, the consistency loss of COLA is relatively high, as shown in Table 3a. However, in Table 3b we see that we find a very similar solution to the one found by HOLA.

For a look-ahead rate of 1.0, COLA's consistency loss is relatively high and as such might not constitute a solution to the consistency problem. We leave it to future work to find methods that obtain better solutions. Nonetheless, the consistency loss for COLA is much lower than the losses for HOLA, as seen in Table 3a. We see in Figure 3e that COLA does not learn a pure Tit-for-Tat strategy but achieves close to the optimal loss, whereas HOLA's loss becomes worse and more stochastic, corresponding roughly to a fully random policy.

7 Conclusion and Future Work

In this paper we cleared up the relation between the Competitive Gradient Descent and LOLA algorithms. We also showed that iLOLA solves part of the consistency problem of LOLA. We introduced a new method, called Consistent LOLA, that finds consistent solutions without requiring many recursive computations like iLOLA . It was commonly believed that the inconsistency lead to arrogant behaviour and lack of preservation for SFPs. Empirically, we show that even with consistency, opponent shaping behaves arrogantly, pointing towards a fundamental open problem for the method.

Furthermore, we empirically investigated the consistency behavior of higher-order LOLA and COLA and found that it does not converge in each hyperparameter regime, even for low-dimensional games with polynomial losses.

This work opens more questions for future work than it answers. Some fundamental questions are the existence and uniqueness of the COLA equations and their relationship to learning outcomes.

Additional work is needed to scale COLA to large settings such as GANs or Deep RL, or settings with more than two-players. Another interesting axis is accounting for further inconsistent aspects of LOLA as identified in Letcher et al. (2019b) and address them via an extension of COLA.

8 ETHICS STATEMENT

General-sum games are an important model for a wide range of real-world scenarios, but have been studied less than zero-sum games. Our work contributes to the foundational study of learning algorithms for these games. It is important to develop learning algorithms that can prevent defect-defect outcomes in situations modeled by the prisoner's dilemma, such as public goods problems. Opponent-shaping methods, as studied in this paper, can achieve this. However, cooperation between the players can also be undesirable, for instance, in markets, where it leads to collusion. Moreover, in the long-term, opponent-shaping could allow agents to manipulate others, including humans. It could thus make sense to also study such negative consequences of opponent-shaping and the limits of their responsible use in the future. However, we believe that at this stage the benefits of furthering our understanding and of developing algorithms that can achieve more cooperative learning outcomes outweigh their downsides.

REFERENCES

Stefano V. Albrecht and Peter Stone. Reasoning about hypothetical agent behaviours and their parameters, 2019.

Robert Axelrod. *The Evolution of Cooperation*. Basic, New York, 1984.

Waïss Azizian, Ioannis Mitliagkas, Simon Lacoste-Julien, and Gauthier Gidel. A tight and unified analysis of gradient-based methods for a whole spectrum of games, 2020.

David Balduzzi, Sebastien Racaniere, James Martens, Jakob Foerster, Karl Tuyls, and Thore Graepel. The Mechanics of n-Player Differentiable Games. *arXiv:1802.05642 [cs]*, June 2018. URL http://arxiv.org/abs/1802.05642. arXiv: 1802.05642.

Andrew Barto and Sridhar Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems: Theory and Applications*, 13, 12 2002. doi: 10.1023/A:1025696116075.

Benjamin Chasnov, Tanner Fiez, and Lillian J. Ratliff. Opponent anticipation via conjectural variations. *Smooth Games Optimization and Machine Learning Workshop at NeurIPS 2020: Bridging Game Theory and Deep Learning*, 2020. URL https://par.nsf.gov/biblio/10223358.

Jakob N. Foerster, Richard Y. Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with Opponent-Learning Awareness. *arXiv:1709.04326 [cs]*, September 2018. URL http://arxiv.org/abs/1709.04326. arXiv: 1709.04326.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Werner Güth, Rolf Schmittberger, and Bernd Schwarze. An experimental analysis of ultimatum bargaining. *Journal of Economic Behavior & Organization*, 3(4):367–388, December 1982. ISSN 0167-2681. doi: 10.1016/0167-2681(82)90011-7. URL https://www.sciencedirect.com/science/article/pii/0167268182900117.

Marc Harper, Vincent Knight, Martin Jones, Georgios Koutsovoulos, Nikoleta E. Glynatsi, and Owen Campbell. Reinforcement learning produces dominant strategies for the iterated prisoner's dilemma. *PLOS ONE*, 12(12):e0188046, Dec 2017. ISSN 1932-6203. doi: 10.1371/journal.pone.0188046. URL http://dx.doi.org/10.1371/journal.pone.0188046.

He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé III au2. Opponent modeling in deep reinforcement learning, 2016.

Joseph Henrich, Robert Boyd, Samuel Bowles, Colin Camerer, Ernst Fehr, and Herbert Gintis. Foundations of Human Sociality: Economic Experiments and Ethnographic Evidence From Fifteen Small-Scale Societies. In *American Anthropologist - AMER ANTHROPOL*, volume 108. January 2006. ISBN 978-0-19-926205-2. doi: 10.1093/0199262055.001.0001. Journal Abbreviation: American Anthropologist - AMER ANTHROPOL.

Adrian Hutter. Learning in two-player games between transparent opponents, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

King Lee and Louis K. *The Application of Decision Theory and Dynamic Programming to Adaptive Control Systems*. Thesis, September 1967. URL https://macsphere.mcmaster.ca/handle/11375/18669. Accepted: 2015-12-21T20:09:58Z.

Alistair Letcher, David Balduzzi, Sebastien Racaniere, James Martens, Jakob Foerster, Karl Tuyls, and Thore Graepel. Differentiable Game Mechanics. *arXiv:1905.04926 [cs, stat]*, May 2019a. URL http://arxiv.org/abs/1905.04926. arXiv: 1905.04926.

Alistair Letcher, Jakob Foerster, David Balduzzi, Tim RocktÃ€schel, and Shimon Whiteson. Stable Opponent Shaping in Differentiable Games. *arXiv:1811.08469 [cs]*, January 2019b. URL http://arxiv.org/abs/1811.08469. arXiv: 1811.08469.

Eric V. Mazumdar, Michael I. Jordan, and S. Shankar Sastry. On Finding Local Nash Equilibria (and Only Local Nash Equilibria) in Zero-Sum Games. *arXiv:1901.00838 [cs, math, stat]*, January 2019. URL http://arxiv.org/abs/1901.00838. arXiv: 1901.00838.

Richard Mealing and Jonathan L. Shapiro. Opponent modeling by expectation–maximization and sequence prediction in simplified poker. *IEEE Transactions on Computational Intelligence and AI in Games*, 9(1):11–24, 2017. doi: 10.1109/TCIAIG.2015.2491611.

Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The Numerics of GANs. *arXiv:1705.10461 [cs]*, June 2018. URL http://arxiv.org/abs/1705.10461. arXiv: 1705.10461.

Hessel Oosterbeek, Randolph Sloof, and Gijs van de Kuilen. Cultural Differences in Ultimatum Game Experiments: Evidence from a Meta-Analysis. *Experimental Economics*, 7(2):171–188, June 2004. ISSN 1573-6938. doi: 10.1023/B:EXEC.0000026978.14316.74. URL https://doi.org/10.1023/B:EXEC.0000026978.14316.74.

Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. The MIT Press, 1994. ISBN 0262150417.

Sébastien Racanière, Theophane Weber, David Reichert, Lars Buesing, Arthur Guez, Danilo Jimenez Rezende, Adrià Puigdomènech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, Razvan Pascanu, Peter Battaglia, Demis Hassabis, David Silver, and Daan Wierstra. Imagination-augmented agents for deep reinforcement learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/9e82757e9a1c12cb710ad680db11f6f1-Paper.pdf.

Alan G. Sanfey, James K. Rilling, Jessica A. Aronson, Leigh E. Nystrom, and Jonathan D. Cohen. The Neural Basis of Economic Decision-Making in the Ultimatum Game. *Science*, 300(5626): 1755–1758, June 2003. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1082976. URL https://science.sciencemag.org/content/300/5626/1755. Publisher: American Association for the Advancement of Science Section: Report.

Florian Schäfer and Anima Anandkumar. Competitive Gradient Descent. *arXiv:1905.12103 [cs, math]*, June 2020. URL http://arxiv.org/abs/1905.12103. arXiv: 1905.12103.

Florian Schäfer, Anima Anandkumar, and Houman Owhadi. Competitive mirror descent, 2020.

J. Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In J. A. Meyer and S. W. Wilson (eds.), *Proc. of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, pp. 222–227. MIT Press/Bradford Books, 1991.

Gabriel Synnaeve and Pierre Bessiere. A Bayesian Model for Opening Prediction in RTS Games with Application to StarCraft. In *Computational Intelligence and Games*, pp. 000, Seoul, South Korea, August 2011. URL https://hal.archives-ouvertes.fr/hal-00607277.

Ben G. Weber and Michael Mateas. A data mining approach to strategy prediction. In *2009 IEEE Symposium on Computational Intelligence and Games*, pp. 140–147, 2009. doi: 10.1109/CIG.2009.5286483.

Ying Wen, Yaodong Yang, Rui Luo, Jun Wang, and Wei Pan. Probabilistic recursive reasoning for multi-agent reinforcement learning, 2019.